CHAPTER

Usage Research Data Modeling

9

Highlights

- How-to suggestions for modeling.
- User work roles.
- User personas.
- The flow model.
- The task structure model: The hierarchical task inventory.
- Task sequence models.
- The artifact model.
- Physical work environment model.
- The information architecture model.
- The social model.
- Hybrid models.

9.1 INTRODUCTION

9.1.1 You Are Here

We begin each process chapter with a "you are here" picture of the chapter topic in the context of The Wheel, the overall UX design lifecycle template (Fig. 9-1). Within the Understand Needs UX design lifecycle activity, this chapter is about the data modeling subactivity in which you represent the usage research data you elicited in Chapter 7 in simple models.

9.1.2 What Are Usage Research Data Models and How Are They Used

Modeling, a basic life skill (Section 2.4.5), is a way to organize some kinds of raw usage research data into representations to inform UX design. Each model provides a different perspective into the overall picture of work practice. Models use abstraction (another life skill) to boil things down to the essence and turn usage research data into actionable items for design.

Abstraction

The process of removing extraneous details of something and focusing on its irreducible constructs to identify what is really going on, ignoring everything else (Section 14.2.8.2).

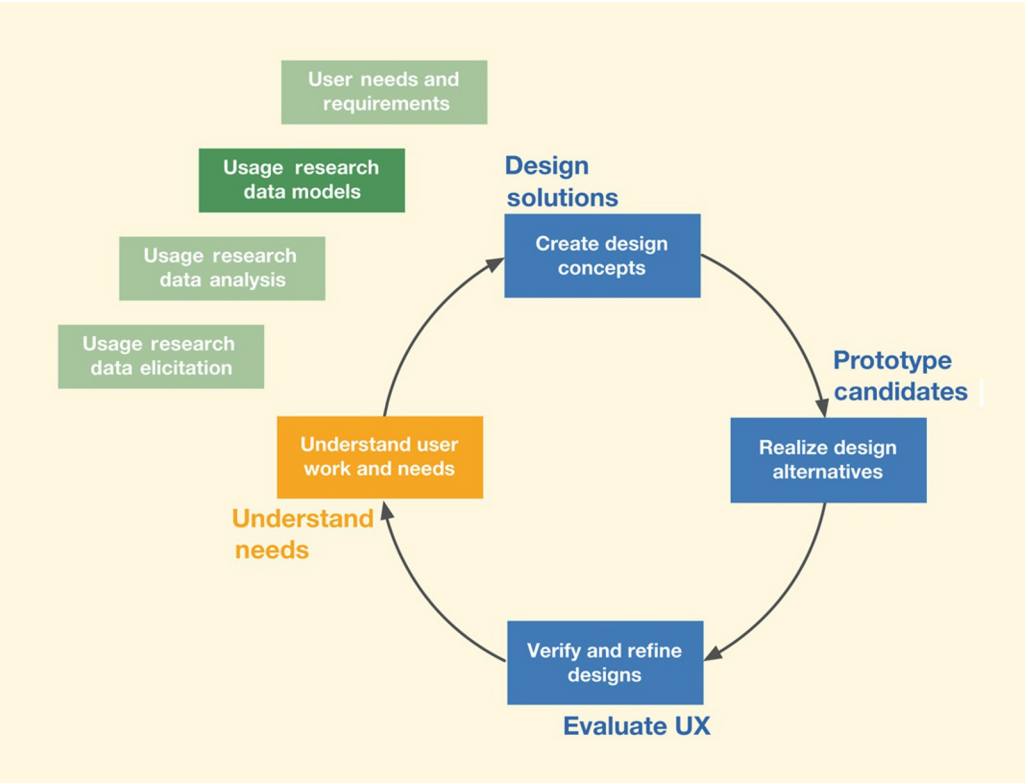

Fig. 9-1

You are here in the subactivity of constructing usage research data models within the Understand Needs lifecycle activity, in the context of The Wheel, the overall lifecycle process.

Usage research models also provide some boundaries and checklists of things that have to be included in the design. Most of all, the models offer an immersive mental framework of design parameters.

9.1.3 Kinds of Data Models

Not every data model is appropriate for every project. In this chapter, we present the models in an order that approximately represents importance and frequency of use:

- The most commonly needed models—in almost every project you should make these models: User work role model and flow model.

- If there is a broad range of user characteristics for any given user work role, consider making a user persona.
- If there are a large number of different user tasks, organize them with a task structure model, such as the hierarchical task inventory.
- If some user tasks are a bit complex, describe them with task sequence models.
- If the work practice is artifact-centric, describe them with an artifact model.
- If the work practice is influenced by physical layout, describe that with a physical work environment model.
- If the data and information users need to store, retrieve, display, and manipulate are complex, represent it with an information architecture model.
- The social mode is the least commonly used model, needed only when social and cultural interactions among the people involved in the work practice are complex and/or problematic.

Artifact model

A representation showing how users employ, manipulate, and share key tangible objects (physical or electronic work practice artifacts) as part of the flow in their work practice (Section 9.8).

9.1.4 Modeling Should Already be Well Established

Starting from the project commission (Section 5.4), basic model-related information is among the earliest things that you learn. The project proposal or business brief that kicks off the project and the early client meetings to define the project all provide basic inputs to at least:

- The user work role model.
- The flow model.
- Task models.

Early usage research, especially data elicitation and analysis, helps fill in gaps and refine concepts. Later usage research is for confirming points and answering questions about the models.

So, by the time you get here, some of your modeling should already be established. We describe it here to get all the modeling methods and techniques in one place rather than as it will occur in your project, interspersed throughout project commission, usage research data elicitation, analysis, and modeling.

9.2 SOME GENERAL "HOW TO" SUGGESTIONS FOR DATA MODELING

9.2.1 How Modeling Can Overlap with Usage Research Data Elicitation and Analysis

For simple things such as data about user work roles and information flow, you can already start sketching out some early models even as you are onsite, eliciting usage research data. This will allow you to pass these early models by users for confirmation (or not).

User work role

Not a person but a work assignment, defined and distinguished by a corresponding job title or work responsibilities. A work role usually involves system usage, but some work roles can be external to the organization being studied (Section 7.5.4.1).

Hierarchical task inventory (HTI)

A hierarchical structural representation of task and subtask relationships for cataloguing and representing the hierarchical relationships among tasks and subtasks that must be supported in a system design (Section 9.6).

Elemental data note

A data note from either usage research or UX evaluation that is brief, clear, concise, and refers to or relates to exactly one concept, idea, fact, or topic (Section 8.2.2).

Barrier

In the context of usage research, a problem that interferes with normal operations of user work practice, impedes user activities or task performance, interrupts work flow or communications, or interferes with the work practice (Section 9.7.2.3).

In usage research analysis (Chapter 8), as you encountered an elemental data note that related to a data model, you either set it aside as an input to modeling or you merged it into that model (Section 8.4). So, for example, if you are now considering a note about a user work role, see if that work role is already fully represented in the user work role model. If not, merge in any new information (e.g., add a new user work role to the list). If the elemental data note mentions a new path of flow of an artifact or information, add an arc and maybe a new node to the flow model (a simple graphical representation of how information and artifacts flow through the system, Section 9.5). If the note mentions a new task for a given user work role, add it to the hierarchical task inventory for that user work role and start a task sequence model for it.

9.2.2 For High Rigor, Maintain Connections to Data Sources

You will not usually require high rigor and traceability of usage research data and model components to original sources. However, if rigor and traceability demands it, you can carry forward source IDs assigned to elemental data to maintain a connection from model components to data sources. This allows you to go back to the original sources of the raw data in question to resolve questions, disagreements, or interpretations of the data.

Example: Tagging Model with Source ID

In the example at the end of Sections 8.2.3 and 8.2.7, we had an elemental data note that described a barrier in the workflow model of the ticket buyer:

> *It is usually difficult to get enough information about events from a ticket seller at the ticket window. [8]*

Where possible, the flow model will also be tagged with "[8]" here, to maintain this "chain of custody" of data source information.

Because most projects do not require maximum rigor of this kind, we do not pursue this source ID tagging in the rest of this chapter.

9.3 THE USER WORK ROLE MODEL

The work role model is one of the most important models and every project needs one. This model, at its base, is a simple representation of user work roles, subroles, and associated user class characteristics. It is essential to identify the operational user work (or play) roles as early in usage research as possible.

9.3.1 What is a User Work Role?

A user work role is not a person, but a work assignment defined by the duties, functions, and work activities of a person with a certain job title or job responsibility, such as "customer" or "database administrator." Job titles themselves, however, don't necessarily make good names for user work roles; you should use names that distinguish them by the kind of work they do.

For example, the MUTTS ticket seller who helps customers buy tickets does entirely different tasks with the system than, say, the event manager who, behind the scenes, enters entertainment event information into the system so that tickets can be offered, purchased, and printed.

Many people can play the same work role. For example, all cashiers at a bank might fall under the same work role, even though they are different people.

A work role can:

- Involve system usage or not.
- Be internal or external to the organization, as long as the job entails participation in the work practice of the organization.

As an example of what kind of raw data to look for in usage research data, any information about the "ticket buyer" in the Ticket Kiosk System should be merged into that user work role model.

Reminder: MUTTS is the old system, using a ticket window and the Ticket Kiosk System is the new system, using public kiosks.

MUTTS

MUTTS is the acronym for Middleburg University Ticket Transaction Service, our running example for most of the process chapters (Section 5.5).

Example: Work Role Identification for MUTTS

The two obvious work roles in MUTTS already mentioned are:

- Ticket buyer, who interacts with the ticket seller to learn about event information and to buy event tickets.
- Ticket seller, who serves ticket buyers and uses the system to find and buy tickets on behalf of ticket buyers.

Among the other roles we discovered early in usage research are:

- Event manager, who negotiates with event promoters about event information and tickets to be sold by the MUTTS ticket office.
- Advertising manager, who negotiates with outside advertisers to arrange for advertising to be featured via MUTTS—for example, ads printed on the back of tickets, posted on bulletin boards, and on the website.
- Financial administrator, who is responsible for accounting and credit card issues.

- Maintenance technician, who maintains the MUTTS ticket office computers, website, ticket printers, and network connections.
- Database administrator, who is responsible for the reliability and data integrity of the database.
- Administrative supervisor, who oversees the entire MU services department.
- Office manager, who is in charge of the daily MUTTS operation.
- Assistant office manager, who assists the office manager.

9.3.2 Subroles

For some work roles, it can be useful to distinguish subroles defined by different subsets of tasks the work role does. Examples of subroles for the ticket buyer role include student, general public, faculty/staff, and alumni ticket buyers.

9.3.3 Mediated Work Roles

Some "users" serve roles that do not use the system directly but still play a major part in the workflow and usage context. We call these users "mediated users" because their interaction with the system is mediated by direct users. Cooper (2004) calls them "served users" and they still have true work roles in the enterprise and are true stakeholders in the system.

Mediated roles are often customers and clients of the enterprise on whose behalf direct users such as clerks and agents conduct transactions with the computer system. They might be point-of-sale customers at a retail outlet or customers needing services at a bank or an insurance agency. The working relationship between the mediated users and the agent is critical to the resulting user experience. The ticket-buyer role for MUTTS is a prime example of a user role whose interaction with the computer system is mediated (by the ticket seller).

Work activity note

A brief, clear, concise, and elemental (relating to exactly one concept, idea, fact, or topic) statement used to document a single point about the work practice as synthesized from raw usage research data (Section 8.1.2).

Exercise 9-1: Identifying User Work Roles for Your Product or System

Goal: Get a little practice at identifying work roles from your work activity notes.

Activities: By now you should be pretty certain about the work roles for your system. Using your user-related work activity notes, identify and list the major work roles for your product or system.

For each role, add explanatory notes describing the role and add a description of the major task set that people in that role would be expected to perform.

Deliverables: A written list of work roles you identified for your system, each with an explanation of the role and a brief high-level description of the associated task set.
Schedule: A half hour should do it.

9.3.4 User Class Definitions

A user class for a work role or subrole is defined by a description of relevant characteristics of the potential user community that can perform that role. Every work role and subrole will have at least one user class.

User class definitions can be defined in terms of such characteristics as demographics, skills, knowledge, and special needs. Some specialized user classes, such as "soccer mom," "yuppie," "metrosexual," or "elderly citizen," may be dictated by marketing (Frank, 2006).

As an example of a user class, the set of people in the town-resident subrole of the ticket-buyer role in the new Ticket Kiosk System might be expected to include inexperienced (first-time) users from the general public. Another user class for this work role could be senior citizens with limited motor skills and some visual impairment.

The characteristics used to distinguish user classes can involve almost any credential or qualification that describes attributes needed to perform the corresponding work role.

9.3.4.1 Knowledge- and skills-based characteristics

User class characteristics related to knowledge and skills include:

- Background, experience, training, education, and/or skills expected in a user to perform a given work role. For example, a given class of users must be trained in retail marketing and must have five years of sales experience.
- Knowledge of computers—both in general and with respect to specific systems.
- Knowledge of the work domain—knowledge of and experience with the operations, procedures, and semantics of the various aspects of the application area the system being designed is trying to address.

For example, a medical doctor might be an expert in domain knowledge related to an MRI system, but may have only novice-to-intermittent knowledge in the area of related computer applications. In contrast, an administrator in the hospital may have little overall domain knowledge about MRI systems, but may have more complete knowledge regarding the day-to-day use of related computer applications.

Some knowledge- and skills-based characteristics of user class definitions can be mandated by organizational policies or even legal requirements, especially for work roles that affect public safety.

9.3.4.2 Physiological characteristics

User class characteristics related to physiological characteristics include:

- Impairments, limitations, and other ADA-related considerations.
- Age. If older adults are expected to take on a given work role, they may have known characteristics to be accommodated in design. For example, they can be susceptible to sensory and motor limitations that come naturally with age.

Physiological characteristics are certainly where accessibility issues can be found. Within work roles, you may also find subclasses of users based on special characteristics such as special needs and disabilities.

Example: User Class Definitions for MUTTS

Ticket seller. Minimum requirements might include point-and-click computer skills. Probably some simple additional domain-specific training is necessary. When we interviewed ticket sellers at MUTTS, we discovered that they did have a manual explaining the job responsibilities, but over time it had become outdated and eventually was lost.

Because ticket sellers are often hired as part-time employees, there can be considerable turnover. So, as a practical matter, much of the ticket seller training is picked up as on-the-job training or while "apprenticing" with someone more experienced, with some mistakes occurring along the way. This variability of competence in the work role, which is the main interface with the public, is not always the best for customer satisfaction, but there does not seem to be a way around it at MUTTS.

Exercise 9.2: User Class Definitions for Your Product or System

Goal: Get practice in defining user classes for work roles, similar to the example above.

Activities: Using your user-related usage research data notes, create a few user class definitions to go with the work roles you identified in the previous exercise. Describe the characteristics of each user class.

Deliverables: A few user class definitions to go with the work roles identified.

Schedule: About 30–45 minutes should be enough to get the most out of this assignment.

9.3.5 Post the Work Role Modeling Results

Post a visual representation of the updated and refined work roles, subroles, and user classes in a central location within the design studio, so everyone can refer to it during design.

9.4 USER PERSONAS

A persona, or user persona, is a narrative description of a specific design target of a UX design for a user in one work role. A persona is a hypothetical but specific "character" in a specific work role. As a technique for making users real to designers, a persona is a story and description of a realistic individual who has a name, a life, and a personality, allowing designers to limit design focus to something very specific.

User personas are a user model that is very closely tied to design. We don't make personas for users of the old, existing product or system because we use them to guide the design for the new system. However, they are introduced here with the user models because they are derived from usage research data about users.

9.4.1 What Are User Personas?

First of all, it's important to note that each persona we make applies to exactly one user work role. So you can make a persona for each user work role or just for the important ones.

You can't make a single design that is the best for everyone who might fill the corresponding work role. If you try, you will often end up with a design that is so general it doesn't work well for anyone. So it has been suggested that you create a persona as a specific design target.

A persona is not an actual user, nor necessarily a typical user, and certainly not an average of users. Rather, a persona is a pretend user or a "hypothetical archetype" (Cooper, 2004) person that must be served by the design. Each persona is a single "user" with very concrete characteristics, representing a specific person in a specific work role with specific user characteristics.

9.4.2 Extracting Data for Personas

During usage research analysis, as you look at each work activity note, if it is about a particular user as a person and says anything about their personality and habits and how that person uses the product or system, it is a good candidate for consideration as an input to a user persona model.

Here are a few example elemental data notes that are candidates for inputs to user personas in the TKS:

- I usually work long hours in the lab on the other side of campus.
- I like classical music concerts, especially from local artists.
- I love the sense of community in Middleburg.
- Sometimes I need to buy a set of tickets with adjacent seating because I like to sit with friends.
- I like to buy MU football game tickets in a group so I can sit with my friends.

The main discussion about constructing personas and using them to guide design is in Chapter 14 on generative design.

9.4.3 A Preview of How to Create Personas

For any given work role, personas are defined by user goals arising from their subroles and user classes. Different subroles and associated user classes have different goals, which will lead to different personas and different designs.

Starting with user research data, you first create multiple candidate personas for each user work role. Each persona is a description of a specific individual who is given a name, a life, a personality, and a profile as a person, especially with respect to how they use the new product or system. Making personas precise and specific is paramount. Specificity is important because that is what lets you exclude other cases when it comes to design.

Then the hard part is to choose one to design for. Through a process to be described in the design chapters (Part 3), you will choose one of those personas as the one primary persona, the single best design target, the persona to which the design will be made specific. The trick is to make the target design to be *just right* for your chosen persona and to *suffice* for the others. This part is deferred until the design chapters.

Example: Personas for the Ticket Kiosk System

An example of a persona for the student subrole in the Ticket Kiosk System could be that of Jane, a biology major who is a second-generation MU attendee and a serious MU sports fan with season tickets to MU football. Jane is a candidate to be the primary persona because she is representative of most MU students when it comes to MU "school spirit." Another persona, Jeff, who is a music major interested in the arts, is also an important one to consider to add breadth to the design.

Exercise 9.3: Early Sketch of a User Persona

Goal: Get some experience at writing a persona similar to the example above.
Activities: Select an important work role within your system. It is best if at least one user class for this work role is broad with the user population coming from a large and diverse group, such as the general public. Using your user-related usage research data, create a persona, give it a name, and get a photo to go with it. Write the text for the persona description.
Schedule: You should be able to do what you need to learn from this in about an hour.

9.5 THE FLOW MODEL

The flow model is one of the most important models and every project needs one.

9.5.1 What Is a Flow Model?

The flow model, at its base, is a simple graphical representation of how information and artifacts flow through the system as it is used. It's essential to identify the basic system flow as early in usage research as possible. A flow model gives you an overview of how information, artifacts, and work products flow among user work roles and parts of the product or system as the result of user actions. For example, how does a song or piece of music flow as it is purchased, downloaded from the Internet, and loaded or synchronized to a personal device?

A flow model is a bird's-eye picture of the work domain, its components, and interconnections among them. It's a high-level view of how users in each work role and other system entities interact and communicate to get work done. A flow model is especially about how work gets handed off between roles; the places where things are most likely to fall through the cracks. As Beyer and Holtzblatt (1997, p. 236) put it: "The system's job is to carry context between roles."

9.5.2 Central Importance of the Flow Model

Because the flow model is a unifying representation of how the system fits into the workflow of the enterprise, it is important to understand it and get it established as early as possible. Along with the user work role model, the flow model is the centerpiece of immersion in your UX design studio. Even though your early usage research data will be incomplete and not entirely accurate, you can refine it as a clear picture of the work domain, system, and users slowly emerges. If necessary, you should go back to your users and ask them to verify the accuracy and completeness of the flow model.

Immersion

A form of deep thought and analysis of the problem at hand—to "live" within the context of a problem and to make connections among the different aspects of it (Section 2.4.7).

9.5.3 How to Make a Flow Model

Starting very early in the UX lifecycle:

- Draw the evolving flow model diagram as a graph of nodes and arcs.
- Start by drawing people icons, labeled with the work roles, as nodes.
- Include roles external to the organization.
- Add nodes for other entities, such as a database into which and from which anything related to the work practice can flow.
- Draw directed arcs (arrows) representing flow, communication, and coordination necessary to do the work of the enterprise, between nodes.
- Label the arcs with what (e.g., artifact, information) is flowing and by what medium (e.g., email, phone calls, letters, memos, and meetings).
- In usage research analysis, as you encounter elemental notes that describe how work flows in the organization, set them aside as inputs to the flow model or merge them directly into the evolving flow model.

Flow model

A simple graphical diagram giving a big picture or overview of work by representing how information, artifacts, and work products flow among work roles and system components within the work practice of an organization (Section 9.5).

Flow models include non-UI software functionality, too, when it is part of the flow; for example, the payroll program must be run before printing and then issuing paychecks. If you make a flow model of how a website is used in work practice, do not use it as a flowchart of pages visited. However, it should represent how information and commands flow among the sites, pages, backend content, and users to carry out work activities.

Sometimes you have to make your flow model very detailed to get at important specifics of the flow and to unearth important questions regarding who does what in depth.

Post a large image of the flow model prominently in your UX design studio as a central part of immersion.

Example: Sketching the Flow Model for MUTTS

The simple early MUTTS flow model of Fig. 9-2 is centered on the ticket-buying activity between the ticket seller and ticket buyer roles, based on sketches made in data elicitation and related elemental data notes encountered in usage research analysis.

Interaction between the ticket buyer and ticket seller might begin with a question about what events are available. The ticket seller then sends a suitable request for this information to the event database and the information in a response flows back to the ticket seller, who then tells the ticket buyer. After one

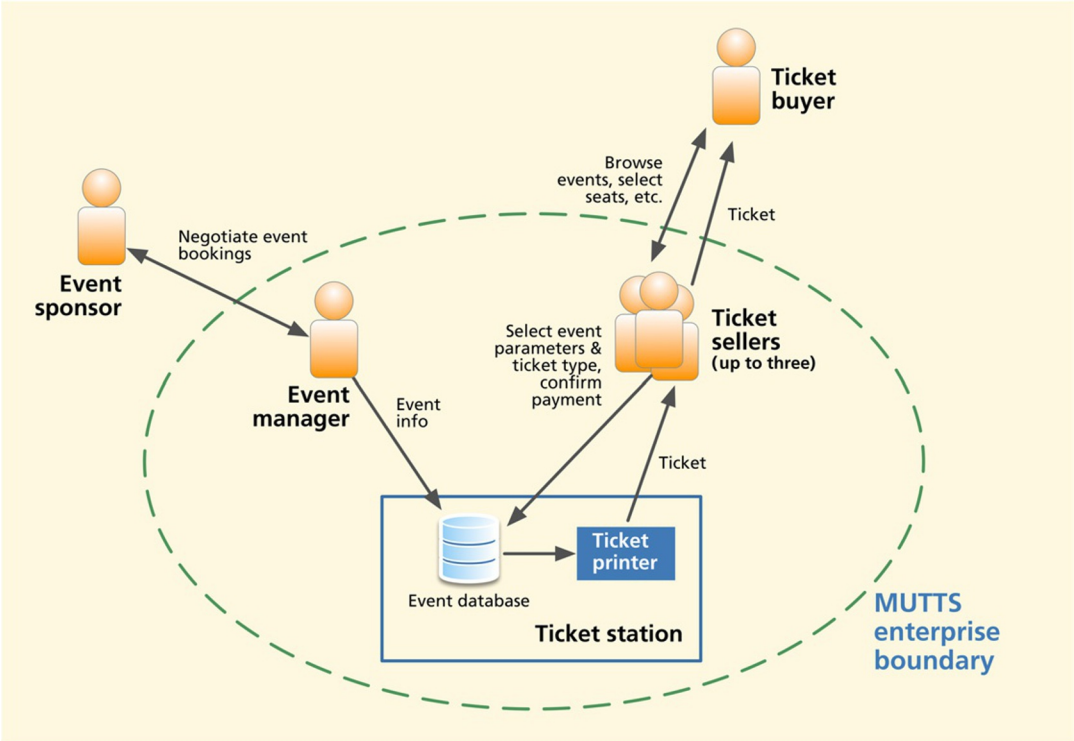

Fig. 9-2

Simple early flow model for the existing MUTTS.

or more such interactions to establish what tickets are available and which ones are desired by the ticket buyer, a ticket purchase request flows from ticket buyer to ticket seller and to the event database. The transaction is consummated with payment and the request then flows to the ticket printer. Printed physical tickets flow to the ticket seller and are then given to (flow to) the ticket buyer.

Example: Extending the MUTTS Flow Model

As more elemental data notes related to the flow of the ticket-buying process emerge in elemental data notes during usage research analysis, we can extend and refine the MUTTS flow model. For instance, in Fig. 9-3 we added an online ticket source, Tickets4ever.com, partnering with MUTTS.

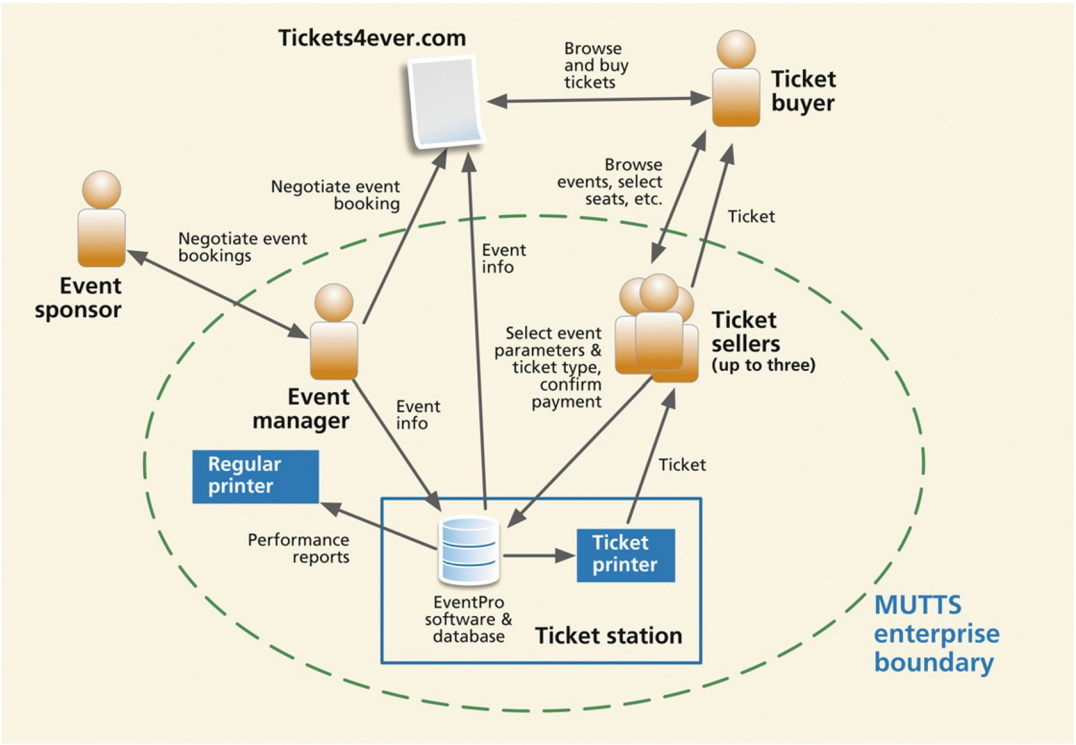

Fig. 9-3

A further step of the flow model sketch of the MUTTS system, showing Tickets4ever.com as a node.

Eventually, the flow model for MUTTS evolved into a rather full diagram, as shown in Fig. 9-4.

Note interactions among roles not involved directly in ticket buying or selling, such as friends and/or family of the ticket buyer in the upper right-hand corner of the diagram, standing there with the ticket buyer or on the cell phone, communicating about choices of events, dates, seats, and prices.

Exercise 9.4: Creating a Flow Model for Your Product or System

Goal: Get practice in making an initial flow model sketch for the work practice of an organization.

Activities: For your target system, sketch out a flow model diagram in the same style as our flow model sketch for MUTTS showing work roles, information flow,

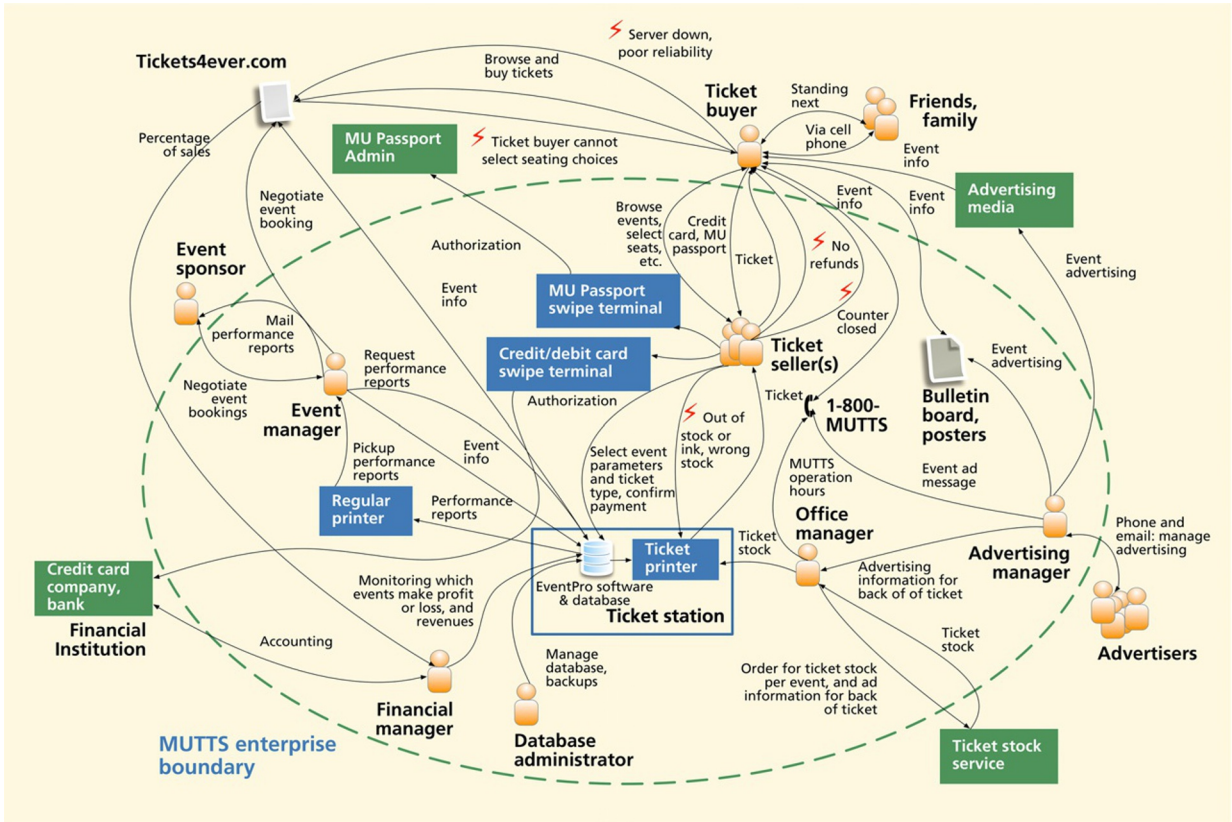

Fig. 9-4

Flow model of our version of MUTTS.

information repositories, transactions, etc. Draw on your raw usage research data and construct a representation of the flow of data, information, and work artifacts.

Start with representing your work roles as nodes, then add in any other nodes for databases and so on.

Label the arcs to show all flow, including all information flow and flow of physical artifacts:

- Show all communication, including direct conversations, email, phones, letters, memos, meetings, and so on.
- Show all coordination necessary to do the work of the enterprise. Include both flow internally within the enterprise and flow externally with the rest of the world.

If you do not have enough usage research data from your limited data elicitation exercise, look up other comparable similar business practices to make this work.
Deliverables: A one-page diagram illustrating a high-level flow model for the existing work process of your target product or system.
Schedule: Given a relatively simple domain, we expect this exercise to take about an hour.

Pervasive information architecture

A structure for organizing, storing, retrieving, displaying, manipulating, and sharing information that provides ever-present information availability spanning parts of a broad ecology (Sections 12.4.4 and 16.2.3).

Ecology

In the setting of UX design, the ecology is the entire set of surrounding parts of the world, including networks, other users, devices, and information structures, with which a user, product, or system interacts (Section 16.2.1).

9.5.4 The Customer Journey Map, a Kind of Flow Model

In Section 7.5.4.2, we described collecting data about extended usage by "shadowing" users and documenting the customer journey, "the product of an interaction between an organization and a customer over the duration of their relationship."[1] Here in data modeling, this kind of data can be represented by what is sometimes called a customer journey map.

As a specific activity model of a different kind of flow in usage, a customer journey map is a "map" of how a customer or user experiences a product or service over time and through space. It is definitely an ecological view of user experience and can often involve pervasive information architecture.

The customer or user passes through and interacts with the ecology over time and through space. A customer journey map tells a story of usage to clients and helps UX designers understand their special work practice and needs.

9.6 TASK STRUCTURE MODELS—THE HIERARCHICAL TASK INVENTORY (HTI)

9.6.1 The Task Models

Task models (task structure and task sequence models) represent what users do or can do. The primary task structure model is the hierarchical task inventory.

The task models are essential for informing UX design. When the elemental data note you are considering in usage research analysis mentions a user task or feature, it is a good candidate to incorporate into a task model.

9.6.2 Benefits of a Task Structure Model

If your product or system has a large number of user tasks in many different categories for many different user work roles, a task structure model such as the hierarchical task inventory (HTI) is a good way to organize the tasks. Task structure models are used to catalog the tasks and subtasks that must be

[1] https:/en.wikipedia.org/wiki/Customer_experience

supported in the system design. Like functional decompositions, task inventories capture hierarchical relationships among the tasks and subtasks.

Task structure models:

- Represent what user tasks and actions are possible in the work practice and work environment, using the system or not.
- Are essential for informing UX design, telling you what tasks (and functionality) you have to design for in the system.
- Serve as a checklist for completeness in the emerging design (Constantine & Lockwood, 1999, p. 99).

A hierarchical task inventory has other advantages, too, with respect to later organizing and managing user stories and as a guide for creating a complete set of user stories as requirements.

9.6.3 Tasks versus Functions

Informally, we may use the terms "task" and "function" more or less interchangeably when talking about the features of a system. But, when we wish to avoid confusion, we use the term "task" to refer to things a user does and the term "function" to things the system does.

When the point of view is uncertain, we sometimes see a reference to both. For example, if we talk about a situation where information is "displayed/viewed," the two terms represent two viewpoints on the same phenomenon. It is clear that "display" is something the system does and "view" is something the user does, as the user and system collaborate to perform the task/function. Within usage research analysis, of course, the user (or task) viewpoint is paramount.

9.6.4 Create an HTI Model

A simple task structure can be easily represented as a hierarchical (indented) list of tasks and subtasks. More complex task structures are best represented in an HTI diagram.

Hierarchical task inventories can be constructed top-down, bottom-up, or both. Large, more general tasks are decomposed into smaller, more specific, and more detailed tasks.

9.6.5 Hierarchical Relationships

Location within the diagram indicates a hierarchical relationship between tasks and subtasks. If task A is immediately above task B in an HTI diagram, as in Fig. 9-5, it means that task B is a subtask of task A and task A is a supertask of

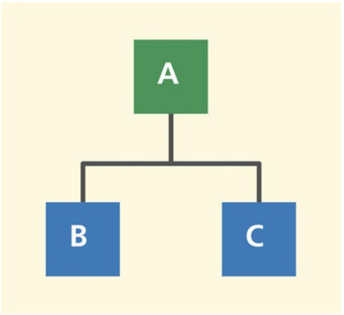

Fig. 9-5
Hierarchical relationship of task A, the supertask, and tasks B and C, subtasks.

task B. The litmus test for this relationship is: If you are doing task B, you are necessarily also doing task A, because task B is part of task A.

The best way to name tasks in this structure is as an <action> <object> pair, such as "add appointment" or "configure parameters," or in an <verb> <adjective> <noun> triple, such as "configure control parameters."

9.6.6 Avoid Temporal Implications

The hierarchical relationship does not show temporal ordering. In Fig. 9-6, we depict an incorrect attempt at a hierarchical relationship because selecting a gear is not part of starting the engine—that is, it fails the litmus test mentioned above.

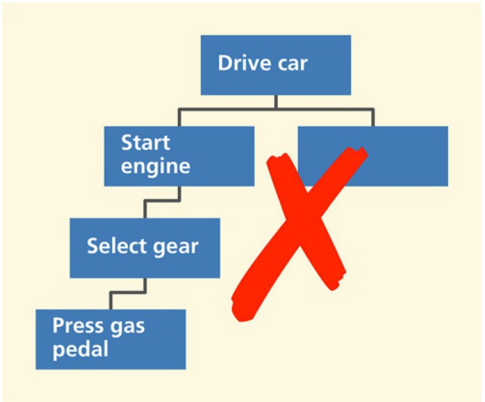

Fig. 9-6
An incorrect hierarchical relationship attempting to show temporal sequencing.

Example: A First-Level HTI Diagram for MUTTS

Starting at the very highest level of tasks for MUTTS, you have the major task sets performed by each of the work roles, such as the financial administrator, the database administrator, the event manager, the advertising manager, and the ticker buyer. Using an "action-object" approach to task naming, these major task sets might be called "manage finances," "manage database," and so on, as shown in Fig. 9-7.

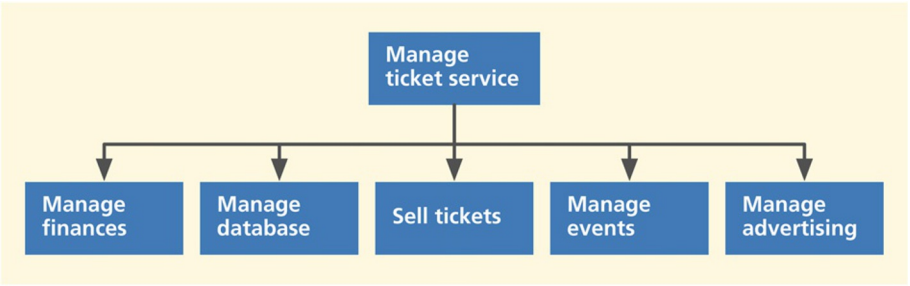

Fig. 9-7
Sketch of the top levels of a possible hierarchical task inventory diagram for MUTTS.

9.6.7 HTI Can Often be Decomposed by User Work Role

The full HTI for a nontrivial system can be enormous. Fortunately, there is a way to control this complexity. Usually you can partition the full HTI by user work roles because the set of tasks performed by one user work role is usually separate from those performed by others. The first level from the top is where this separation by user work roles is seen (Fig. 9-8), resulting essentially in a separate HTI diagram for each user work role.

The team can usually consider each user work role diagram more or less separately in analysis and design. Of course, there are system-spanning issues and parts of the design that involve multiple user work roles, but this imperfect compartmentalization helps a great deal by allowing UX designers to look at the whole system a piece at a time.

The HTI for just the ticket seller role in MUTTS is shown in Fig. 9-9. The "sell tickets" task encompasses all event searching and other subtasks that necessarily go into making a final ticket sale.

Exercise 9.5: HTI for Your Product or System

Goal: Get some practice creating an HTI diagram.
Activities: Using your task-related work activity notes and what you know about your product or system, make a simple HTI diagram for your system.

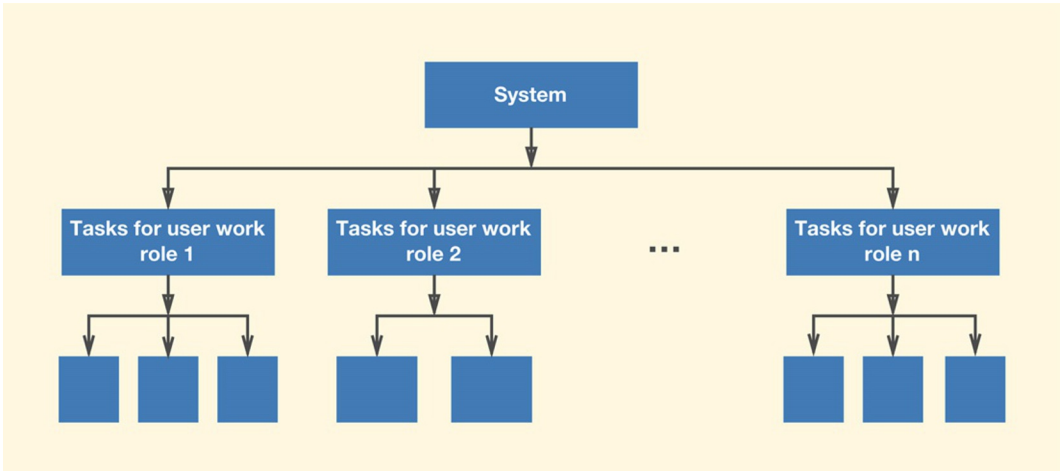

Fig. 9-8

Separation by user work roles at the top of a task hierarchy.

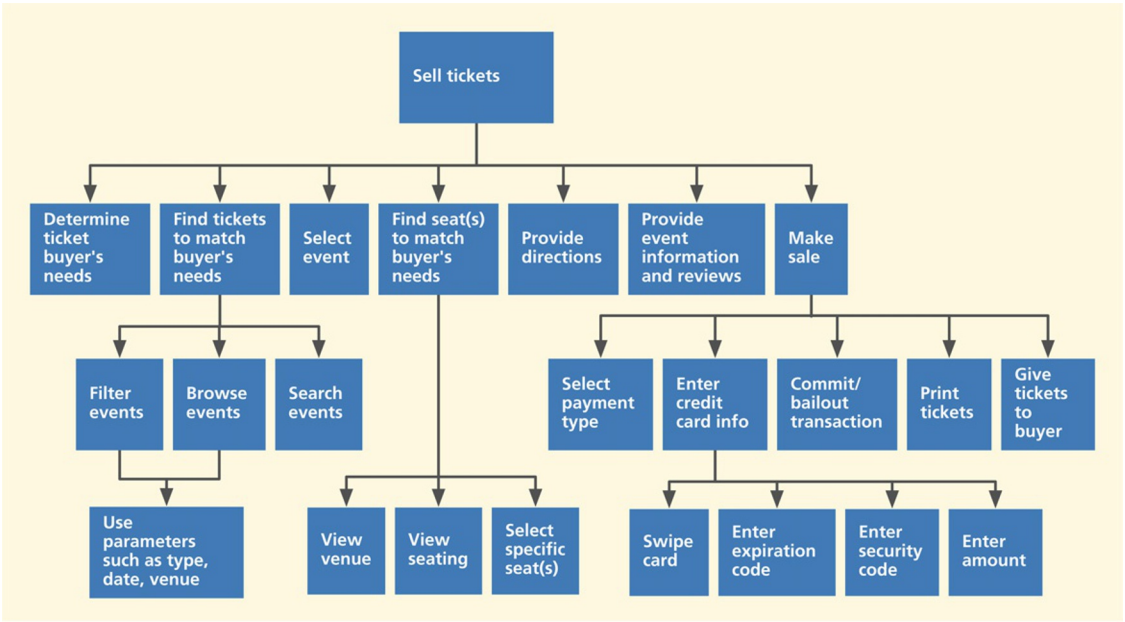

Fig. 9-9

Example HTI for the ticket seller role in MUTTS.

Deliverables: Simple HTI diagram(s) for the product or system of your choice.
Schedule: An hour should be enough to get what you need from this exercise.

9.7 TASK SEQUENCE MODELS

If you have elemental data notes that describe user tasks, you can represent them with a number of different task sequence models.

State diagram (in UX)

A directed graph in which nodes are states which correspond to screens (in the broadest sense), and arcs (or arrows) are transitions between states resulting from user actions or system events. Used in wireflow and wireframe prototypes to show navigation among screens (Sections 9.7.6 and 20.4.4.2).

9.7.1 What Are Task Sequence Models?

Task sequence models are step-by-step descriptions of how users perform tasks with a product or system. User actions and system responses are often separated into dual (or multiple) "swim lanes." Task descriptions have certain "boilerplate" parameters, as described in the subsections below.

Other kinds of task sequence models are used to highlight user workflow. These include state diagrams, which are a kind of hybrid abstraction between a flow model and a task-sequence model that shows user navigation and how information and artifacts in the user workflow are passed among user work roles and other active agents (e.g., a database system).

***Scenario*s.** A scenario is a description of specific people performing work activities in a specific work situation within a specific work context, told in a concrete narrative style as if it were a transcript of a real usage occurrence. Scenarios are deliberately informal, open ended, and fragmentary narrative depictions of key usage situations happening over time.

A *usage scenario* is a description of a way someone uses an existing product or system. A *design scenario* is a description of envisioned usage of a product or system being designed.

Example: Simplest Early Task Sequence Model—Usage Scenario for a Menu Planning Application

Sometimes the earliest effective task sequence model is a usage or design scenario, a narrative description of things users will do with a proposed system as captured during usage research data elicitation. Here is one such example.

This example is taken, with permission, from a graduate-level HCI class student project about a (food) menu management system. As these students said in their report: written at first at a high level, these scenarios were expanded to include more detail and were used to check the emerging design against requirements, identify system states, and to identify tasks and even early prototype interface features.

Scenario 1: Lois's father, who is diabetic, will be moving in with her while her mother is in the hospital. Lois is not accustomed to cooking for someone who is diabetic. Because she is also on a diet, she also wants meals that are low fat. To make things easier, she turns to Menu-Bot, her computerized meal planner, to develop meal plans suitable for both her father and herself.

Lois first creates a new meal plan and sets it up to cover three days. For the first day's breakfast, she decides on a simple meal of coffee, juice, and toast. Because she and her father both work, she decides to prepare a simple seafood salad that they can take with them to work for lunch.

However, for dinner she wants a menu with a main dish, a soup, and two side dishes; a dessert would also be nice. Because she doesn't want to have to wade through all the recipes in Menu-Bot but only those low in sugar and fat, she requests that it provide her with only chicken dishes that are classified as low sugar (suitable for hyperglycemic individuals) and low fat. For the main course, she selects from the set that Menu-Bot recommends: low-fat and low-sugar dishes of asparagus and crab soup, lemon chicken, green beans, and herb-baked potatoes.

Later that evening, returning from work first, Lois's father decides to try his hand at cooking. Having seen Lois use Menu-Bot, he opens up the meal plan she has prepared and selects the evening menu. Not being very experienced at cooking, he lets Menu-Bot instruct him in the steps of food preparation and cooking.

Scenario 2: Bob likes to cook so many different types of food that his kitchen is cluttered with cookbooks. In the past, locating recipes that he is "in the mood for" was a gargantuan feat. Now, however, he has Menu-Bot to help. He is planning a dinner party for six guests and has already made up his mind to serve baked salmon with an almond sauce on wild rice, along with lemon butter squash. So he enters just those dishes directly into the system and gets some good recipes that will go well together. However, he still needs an appetizer and dinner wine, so he lets Menu-Bot recommend them and, seeing that they are good, he adds them to the dinner menu.

Exercise 9-6: Usage Scenarios as Simple Task-Sequence Models for Your Product or System

Goal: Get some practice in writing usage scenarios as early, simple task-sequence models.

Activities: Select one or two good representative task threads for one of your user work roles, for example, the customer.

- Write a couple of detailed usage scenarios, referring to user roles, tasks, actions, objects, and work context.

- Work quickly; you can clean it up as you go.

Deliverables: A few usage scenarios to share and discuss.
Schedule: An hour should be enough time for this one.

9.7.2 Components of Task Sequence Models

9.7.2.1 Task and step goals

A task or step goal is the purpose, reason, or rationale for doing the task or taking the step. Called the user "intent" by Beyer and Holtzblatt (1998), the goal is a user intention, in the sense of being "what the user wants to accomplish" by doing the task.

9.7.2.2 Task triggers

A task trigger (Beyer & Holtzblatt, 1998) or step trigger is something that happens, an event or activation condition, that leads that user to initiate a given task or task step. For example, an incoming phone call leads to filling out an order form. If the user logs into a system, it is because a need arose, maybe the need to enter data from a form.

Triggers are easy to identify in your data notes in usage research analyses. New work arrives in the user's inbox, a new aircraft appears on the air traffic controller's screen, an email request arrives, or the calendar says a report is due soon.

9.7.2.3 Task barriers

Indications of task barriers in your usage research data include user problems and errors that get in the way of successful, easy, and satisfying task or step completion. Task barriers are "pain points" or "choke points" that frustrate users and block productivity or flow. To indicate barriers in our data models, we will use the Beyer and Holtzblatt symbol of a graphical red lightning bolt (), which you should put at the beginning of an indented line explaining the barrier. If the user's reaction or response to a barrier is known through the usage research data, add a brief description of that right after the barrier description among the task steps.

Task barriers are of special interest in UX design because they point out where users have difficulties in the work practice, which in turn are key opportunities for improvement in the design.

9.7.2.4 Information and other needs in tasks

An important component of a task description is the identification of unmet user information and other needs at any step, one of the largest sources of barriers in task performance. The usage research data elicitation and analysis processes and

modeling can help you identify these needs, which you can represent with specific annotations. Just before the step in which the need occurs, add an indented line beginning with a red block "N,"—N—followed by a description of the need.

9.7.3 How to Make a Step-by-Step Task Sequence Description

Task-sequence model

A step-by-step description of how a user might perform a task with a product or system, including task goals, intentions, triggers, and user actions (Section 9.7).

- In either usage research analysis or here in modeling, as you encounter elemental data notes that mention tasks, task steps, or subtasks, merge them into the appropriate evolving task sequence model.
- Sequential steps can be written as an ordered list without the need for flowchart-style arrows to show the flow:
  - Linear lines of text are less cluttered and easier to read.
- Start by showing the most common steps users would take:
  - This is sometimes called the "happy path" or the "go path."
  - This gives a quick and easy-to-understand idea of the basic task without clutter of special cases.

At the beginning, individual task interaction models will be mostly linear paths without branching. Later you can add the most important branching or looping (Section 9.7.4) to cover conditional and iterative cases.

So, for example, an initial task interaction model for an online purchase might not show a decision point where the user can pay with either a credit card or PayPal. It would just be a linear set of steps for the task of buying a ticket with, say, a credit card. Later, a separate linear path for the alternative of paying with PayPal is merged, introducing a decision-making point and branching.

Going from a user story to a related task sequence model to a wireframe design is a very natural path that works well in agile UX design. This is another example of how modeling can be distributed throughout the UX lifecycle.

Example: Rudimentary Task Sequence Model for MUTTS

Consider an extremely simple step-by-step task sequence representation for a MUTTS ticket transaction:

MUTTS Ticket Buyer	**MUTTS Ticket Seller**
1. Wait in line until turn.	2. Greet ticket buyer.
3. Describe event for which tickets are wanted.	4. Look up event in database. How many tickets will be needed?
5. State number of tickets.	6. Look up venue seating chart. Describe available options by seat location and price.
7. Select seats.	8. Calculate and state total cost. How would you like to pay?
9. Pay with credit card.	10. Give tickets, receipt, and card to ticket buyer.

You can see that this is a bare task sequence skeleton. You will add other steps and details as you learn about them.

Example: More Detailed Step-by-Step Task Interaction Model for MUTTS

Task name: Finding entertainment for a given date (performed by ticket seller on behalf of ticket buyer).

Task goal: Helping a ticket buyer choose and buy a ticket for entertainment for this coming Friday night.

Task trigger: Ticket buyer arrives at the MUTTS ticket window on the way home from work on a Thursday evening, thinking ahead to the weekend.

Note: It can help analysis and discussion to number the steps so you can refer to them, as we do in this example.

Ticket Buyer	Ticket Seller
1. Tell ticket seller about general goal of wanting to find an entertainment event for the next night (Friday night).	
2. Ask agent about available types of entertainment.	3. Tell ticket buyer that there are plays, concerts, movies, and sports.
4. Not enough information yet to decide on the category. Ask to see examples of different types.	

Step goal: Consider examples of entertainment events.

5. Ask what events are available for Friday night.

Barrier: Agent sees that the number of results is too large to sort through or tell the customer about.

Response to barrier:

	6. Ask customer how to filter results or narrow it down (e.g., "Tell me more about what you like").
7. Ask about something within reasonable walking distance downtown or near a Middleburg bus stop.	8. Tell about some possibilities.

Task continues:

9. Think about the list of possibilities.

: It is difficult to think about specific events while remembering all the others given orally on the list.

Response to barrier:

10. Make a few sketchy notes by hand.

Trigger: Movies seem attractive to ticket buyer.
Goal: Find a movie to see.

11. Tell agent about switching focus to just movies. 12. Tell agent to use the same criterion about being within reasonable walking distance downtown or near a Middleburg bus stop. 14. Consider possibilities and finds a few he likes. 15. Write choices down on paper.	13. Tell about possibilities.

Trigger for interruption to embedded task: Thinks a friend might also like these movies.
N: Needs to know friend's opinion of the selections.
Goal: Contact a friend to help narrow these last few choices down and pick something together.

16. Ask agent to please wait. 17. Call friend on cellphone. 18. Make choice with friend.

Trigger: Choice made, ready to buy two tickets.
Goal: To buy tickets.

19. Tell agent to buy two tickets to selected movie.	20. Set up transaction in computer. 21. Ask: Cash or credit card?
22. Give agent credit card. 24. Sign credit transaction.	23. Swipe card. 25. Print tickets and receipt. 26. Give printed tickets and return credit card and receipt.

9.7.4 Beyond Linear Task Sequence Models

Branching and looping. Although step-by-step task sequence models are primarily for capturing linear sequences of representative task steps, sometimes you encounter a point in the work practice where there is a choice. You can generalize the task sequence representation by showing this choice using branching, as shown with arrows on the left side of Fig. 9-10.

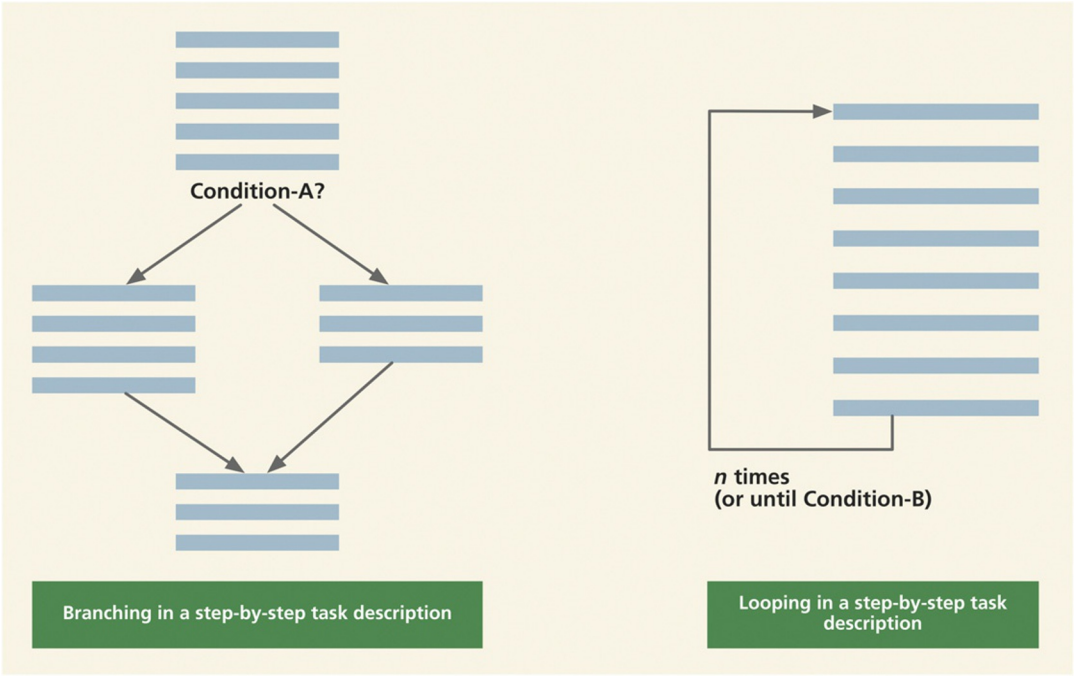

Fig. 9-10

Branching and looping structures within step-by-step task sequence models.

Similarly, if you observe iteration of a set of tasks or task steps, you can represent that as shown on the right side of Fig. 9-10. For sets of steps that are repeated or iterated, note the number of iterations or the condition for termination.

Example: Task Sequence Branching and Looping for MUTTS

In Fig. 9-11, we show a sketch of task sequence representation for selling tickets with MUTTS. Note several instances of looping to iterate parts of the task and, in the bottom box, branching to accommodate two different cases.

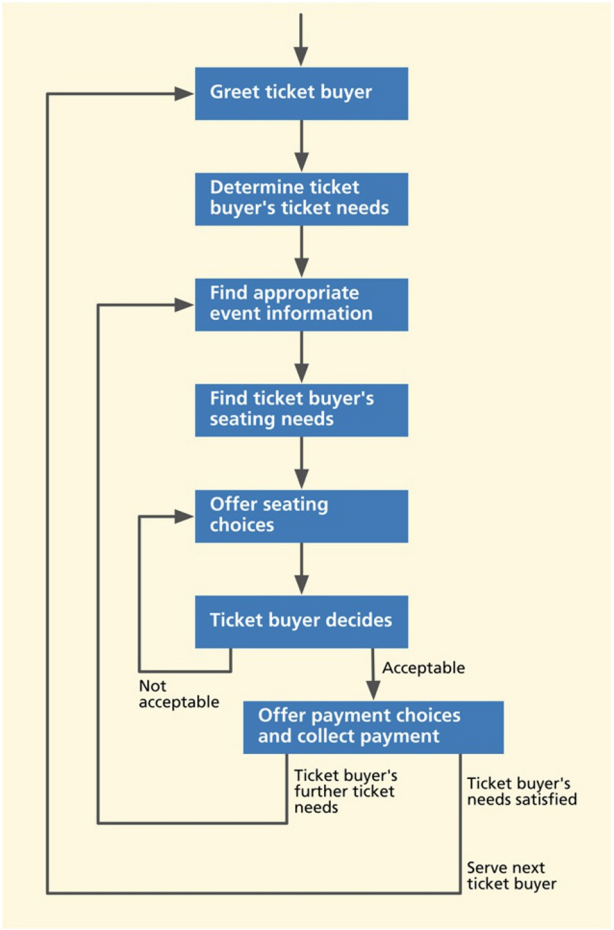

Fig. 9-11
Task sequence branching and looping for MUTTS.

9.7.5 Essential Use Case Task Sequence Models

By combining the best characteristics of step-by-step task descriptions and software use cases, Constantine and Lockwood (1999, p. 100) created essential use cases as an alternative task sequence modeling technique. An essential use case (Constantine & Lockwood, 2003):

- Is a structured narrative.
- Is expressed in the language of users in the work domain.
- Describes a task associated with a single user intention or goal.
- Is abstract.

Naming essential use cases. According to Constantine and Lockwood, each essential use case is named with a "continuing verb" to indicate an ongoing intention, plus a fully qualified object, for example, "buying a movie ticket." Essential use cases capture what users intend to do and why but not how. An example would be searching for a particular entertainment event but nothing about user actions, such as clicking a button.

An essential use case is abstract. The term "essential" refers to abstraction. An essential use case contains only steps that are the essence of the task. The representation is a further abstraction in that it represents only one possible task thread, usually the simplest thread without all the alternatives or special cases. Each description is expressed as a pure work-domain representation, independent of technology or how it looks in the UX design. As an abstraction, an essential use case is a skeleton on which a task description can be woven.

Abstraction

The process of removing extraneous details of something and focusing on its irreducible constructs to identify what is really going on, ignoring everything else (Section 14.2.8.2).

Essential use cases help structure the interaction design around core tasks. These are efficient representations, getting at the essence of what the user wants to do and the corresponding part played by the system.

To illustrate, in Constantine and Lockwood's (2003) ATM example, the user's first step is expressed as an abstract purpose, the "what" of the interaction: "identify self." They do not express it in terms of a "how"; for example, they do not say the first step is to "insert bank card." This is a deceptively simple example of a very important distinction.

Example: Essential Use Case for TKS

Table 9-1 contains an example, cast in the same fashion as Constantine and Lockwood (2003). Notice how these descriptions are abstract.

Table 9-1

Example essential use case: Paying for a ticket purchase transaction (with a credit or debit card)

Ticket Buyer Intention	Ticket Seller and System Responsibility
1. Express intention to make a transaction.	2. Request user to identify self.
3. Identify self.	4. Request to state desired transaction.
5. State desired transaction.	6. Initiate possible negotiation of transaction parameters.
7. Participate in possible transaction negotiations.	8. Summarize transaction and cost.
	9. Request transaction confirmation.
10. Submit confirmation.	11. Conclude transaction.

Note how the abstraction leaves room for design. For example, user identification could be accomplished via a credit card. Transaction confirmation could be submitted in the form of a signature and the transaction could be concluded with a receipt.

For a brief discussion of the roots of essential use cases in software use cases, see Section 11.5.

Task-sequence model

A step-by-step description of how a user might perform a task with a product or system, including task goals, intentions, triggers, and user actions (Section 9.7).

Exercise 9.7: Task Sequence Model for MUTTS Ticket Buying

Goal: Create your own more detailed essential use case model for the ticket-selling task done by the ticket agent in the MUTTS ticket office.

Activities: Select a key ticket-buying agent task and give it a name.

Break one possible task performance thread into steps, including all the steps involving interacting with the customer outside the system, and for each step where appropriate:

- Identify user intent.
- Task triggers.
- Note any possible breakdown points.

Write one out as a task sequence.

Deliverables: A task sequence, written out as an essential use case model.

Schedule: Should take about 30 minutes.

State diagram (in UX)

A directed graph in which nodes are states which correspond to screens (in the broadest sense), and arcs (or arrows) are transitions between states resulting from user actions or system events. Used in wireflow and wireframe prototypes to show navigation among screens (Sections 9.7.6 and 20.4.4.2).

9.7.6 State Diagrams: The Next Step in Representing Task Sequencing and Navigation

A state diagram, a kind of hybrid between a flow model and a task sequence, is often useful for representing details of flow and navigation in the interaction view of design. A state diagram is a form of flow model, used to represent state changes (navigation among screens) in response to user input actions. This is an abstraction to show and understand the main workflow patterns and paths. See also Section 20.4.4.2.

While a state diagram can be used to represent any level of detail, initial state diagrams are most useful for establishing initial wireframes if you stick with the main navigational paths and leave out unnecessary detail, such as error checking, confirmation dialogue, etc. In a transactional system, the flow model can get very complex and edge cases can multiply. An abstract state diagram can help you find the essence of the flow to form the backbone of the simplest version of the wireflow design. When you get to designs of wireframes for system screens, you can think of each screen as being a state in the state diagram. You are designing for things that "live" in that state. To avoid repetition with the full description of state diagrams, we refer you to Section 20.4.4.2 for more about what state diagrams are and how to make and use them.

In early design, your state diagrams can easily be translated into the structure of a wireframe. More details and examples are given in Chapter 20.

9.8 ARTIFACT MODEL

Not all work practice is artifact-centric but, if it is, describe key objects with an artifact model. What artifacts do users employ, manipulate, and share as part of their work practice? Now is the time to take the artifacts (e.g., paperwork) involved in product or system usage that you collected in data elicitation and incorporate them into a simple artifact model. *An artifact (work artifact) is an object, usually tangible, that plays a role in the work flow of a system or enterprise—for example, a printed receipt in a restaurant.*

9.8.1 What's in an Artifact Model?

At this point, your artifact model is probably just a collection of labeled artifacts plus some notes about them. Examples of artifacts include:

- Work practice forms.
- Sketches.
- Props.
- Memos.
- Significant email messages.
- Correspondence templates.
- Product change orders.
- An order form.
- A receipt.
- Paper or electronic forms.
- Templates.
- Physical or electronic entities that users create, retrieve, use, or reference within a task and/or pass on to another person in the work domain.
- Photos (with permission) of the work place and work being performed.
- Other objects that play a role in work performed.

Work artifacts are one of the most important entities that gets passed from one work role to another within the flow model.

Example: Work Artifacts from a Local Restaurant

One of the project teams in our user experience class designed a system to support a more efficient workflow for taking and filling food orders in a local restaurant, which was part of a regional chain. As part of their data elicitation, they gathered a set of paper work artifacts, including manually created order forms, "guest checks," and a receipt, as shown in Fig. 9-12.

Wireflow

A prototype that illustrates navigational flow within an interaction design, representing flow as a directed graph in which nodes are wireframes and arcs are arrows representing navigational flow among the wireframes (Section 20.4.3.1).

Wireframe

A visual template of a screen or webpage design in the interaction perspective, comprised of lines and outlines (hence the name "wire frame"). A skeletal representation of the layout of interaction objects such as tabs, menus, buttons, dialogue boxes, displays, and navigational elements (Section 17.5).

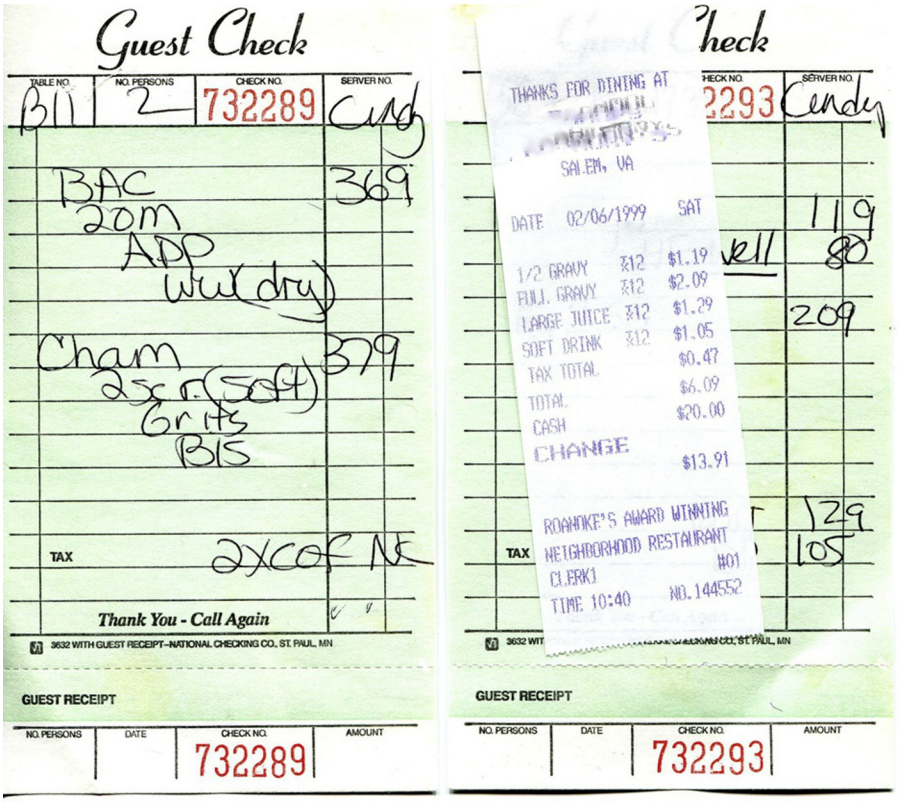

Fig. 9-12

Examples of work artifacts gathered from a local restaurant.

These artifacts are great conversational props as a team interviews those in the different roles that use them. They provide avenues for discussion because almost every restaurant uses these artifacts over and over again. What are things that work with this kind of artifact for order taking? What are some breakdowns? How does a person's handwriting impact this part of the work activity? What is the interaction like between the wait staff and the restaurant's guests? And the interaction between the wait staff and the kitchen staff?

9.8.2 Constructing the Artifact Model

How do you make the artifact model? Well, you now have a collection of artifacts that you gathered in usage research data elicitation, including sketches, copies of paperwork, photographs, and real instances of physical work practice artifacts.

As appropriate, make posters as exhibits ready for discussion and analysis. Attach samples of each artifact and photos of their use to poster paper (e.g., a blank flip chart page). From your data notes, add annotations to these

exhibits to explain how each artifact is used in the work practice. Add stick-on notes associating them with tasks, user goals, and task barriers.

These posters are tangible and visual reminders of user work practice that everyone can walk around, think about, and talk about and, by themselves, are a centerpiece of immersion.

But the real importance of work practice artifacts is in how they tell a story within the overall flow. How seamlessly does a given artifact support the flow? Does it cause a breakdown? Is it a transition point between two systems or roles? In this regard, artifacts are an essential component of the flow model.

Example: Combining the Artifacts Into a Restaurant Flow Model

It is easy to think of artifacts associated with the flow model of a restaurant. The first artifact encountered by a person in the customer work role, delivered by the person in the wait-staff work role, is a menu, used by the customer work role to decide on something to order.

Other usual restaurant work artifacts include the order form on which the wait-staff person writes the original order and the guest check (Fig. 9-12 above), which can be the same artifact or a printed check if the order is entered into a system. Finally, there might be a regular receipt and, if a credit card is used, a credit card signature form and a credit card receipt. Artifacts in restaurants, as they do in most enterprises, are the basis for at least part of the flow model. In Fig. 9-13, you can see how restaurant artifacts help show the workflow from order to serving to concluding payment.

The artifacts, especially when arranged as part of a flow model, can help make a connection from usage research data to thinking ahead about design. For example, the waiting and boredom shown in the figure pose a "barrier" to the customer, represented by a red lightning bolt (). But that waiting time is also a design opportunity. Providing music, news, or entertainment during the wait can alleviate the boredom and gain a competitive advantage in the market.

Example: Artifact Model of Car Keys and Repair Orders in an Auto Repair Shop

As a nonrestaurant example of artifacts in work flow, consider the keys to a customer's car as a central artifact in the work practice of an auto repair facility. First, they may be put in an envelope with the work order, so the mechanic has the keys when needed. If you are working for a project like this and are not able to

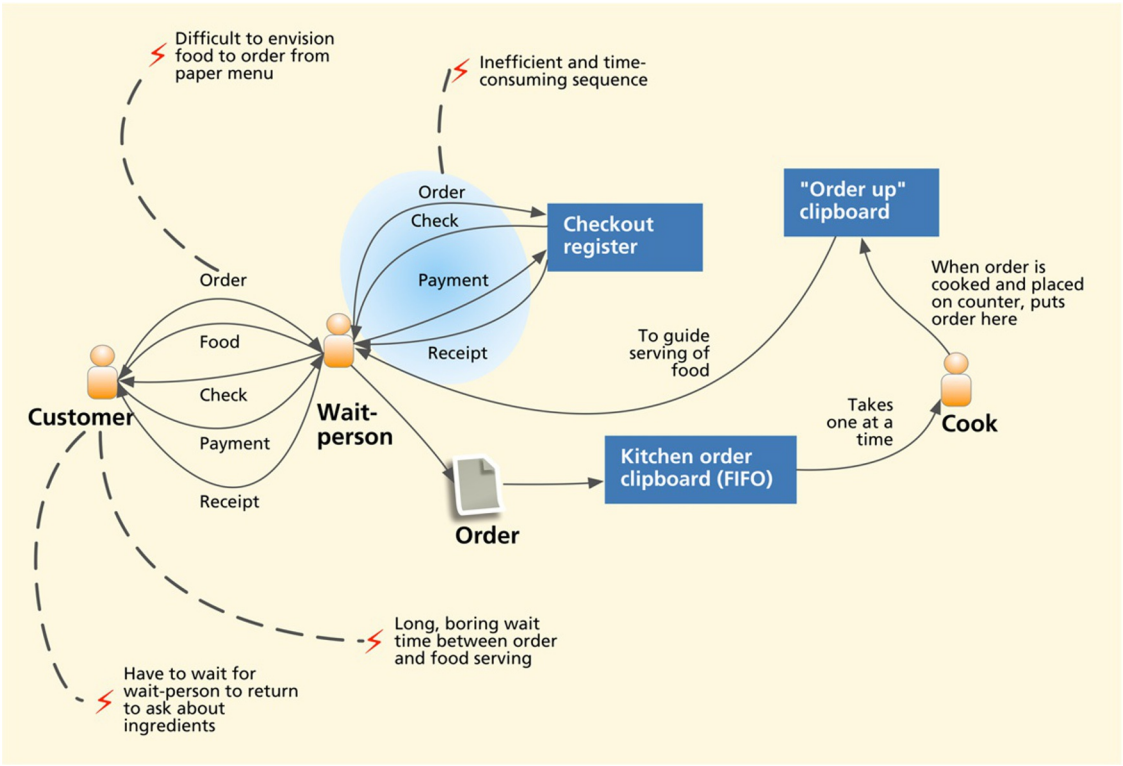

Fig. 9-13

Early sketch of a restaurant flow model with focus on work artifacts derived from the artifact model.

get access to actual keys, improvise. Get a set of old keys that no one needs anymore and use them as a stand-in for the real thing in the artifact model.

After repairs, the keys are hung on a peg board, separate from the repair order until the invoice is presented to the customer and the bill is paid. To support immersion, you should make yourself a mockup of the peg board and a repair order and add them to the growing collection of artifacts in the model.

9.9 PHYSICAL WORK ENVIRONMENT MODEL

Physical models are pictorial representations of the physical layout of work locations, personnel, equipment, hardware, physical parts of the ecology, communications, devices, and databases that are part of the work practice, especially important where the physical layout of entities in the flow matter in the outcome of the work. For

example, the location of desks on a trading floor in a financial institution could be important because of the many exchanges of information that happen through sign language and other verbal communications between traders.

In usage research analysis, as you encounter elemental data notes related to physical workplace layout and how it impacts work practice, merge them into the evolving physical model. Also look for any photos you captured during your elicitation visits. Include sketches, diagrams, and photos of the working environment:

- Physical workspace layout.
- Floor plans (not necessary to scale).
- Where people and important objects stand and where they move to during interaction.
- Locations of:
  - Furniture.
  - Office equipment (telephones, computers, copy machines, fax machines, printers, scanners).
  - Communications connections.
  - Work stations.
  - Points of contact with customers and the public.

As an example, show in a restaurant where the customer tables are and where the wait staff go to submit and pickup orders. Make posters of the diagrams and photos, annotated with notes about the physical layout and any problems or barriers it imposes. Post them in your design studio.

Because a physical model shows the placement and paths of movement of people and objects within this workspace layout diagram, it can be used to assess the proximities of task-related equipment and artifacts and task barriers due to distances, awkward layouts, and physical interference among roles in a shared workspace.

Physical work environment model

Pictorial representation of the milieu in which work gets done, including the physical layout of work locations, personnel, equipment, hardware, physical parts of the ecology, communications, devices, and databases that are part of the work practice (Section 9.9).

Example: How a Friend Used a Physical Model to Help with a House

For example, in the design for her house, a friend worked out a model of workflow within a physical model and found that the traditional American proclivity for putting the clothes washer and dryer in the basement gave a very poor proximity-to-task-association ratio for the task of doing laundry. Enlarging the dressing room and putting the washer and dryer in there improved this ratio enormously.

Similarly, the flow of fresh vegetables from the car to the kitchen led to moving the garage from the basement level to the living floor level (aided by a steep grade). In both cases, the changes brought the physical model elements much closer to their location of use in the design.

Looking further at the veggie flow in the physical model led to an efficient design of a kitchen island as a shared food preparation and cooking area—cleaning at the veggie sink, flowing over to slicing and dicing, and then flowing to sautéing under a bright light and a vent hood.

9.9.1 Include Hardware Design, When Appropriate

In some projects, especially in the commercial product perspective, hardware devices can have enormous importance. For example, if you are designing a new smartphone, this is about considering industrial design, material, production issues, heating, and other physical issues such as interference with any wireless antennas in the device, temperature tolerances, and weather proofing. As another example, for the TKS, hardware concerns might include selecting kiosk locations, printers in kiosks, and susceptibility to vandalism and theft. The design of the kiosk itself will require industrial design skills.

9.9.2 Include Environmental Factors, When Appropriate

When creating physical models, also think of all the physical characteristics of a workplace that can affect work activities and add them as annotations. For example, a steel mill floor is about safety concerns, noise, dust, and hot temperatures—conditions where it is difficult to think or work. A system with a terminal on a factory floor means dirty conditions and no place to hold manuals or blueprints. This may result in designs where the use of audio could be a problem, needing more prominent visual design elements for warnings, such as blinking lights.

Example: Physical Model for MUTTS

In Fig. 9-14, we show the physical model for MUTTS. The center of workflow is the ticket counter, containing up to three active ticket seller terminals. On the back wall, relative to ticket sellers, are the credit card and MU Passport swiping stations. This central ticket-selling area is flanked with the manager's and assistant manager's administrative offices.

Barriers not shown in this figure include a barrier to the ticket buyer lines: At peak times, customers may have to wait in long lines outside the ticket window. The scanner in the manager's office, used to digitize graphical

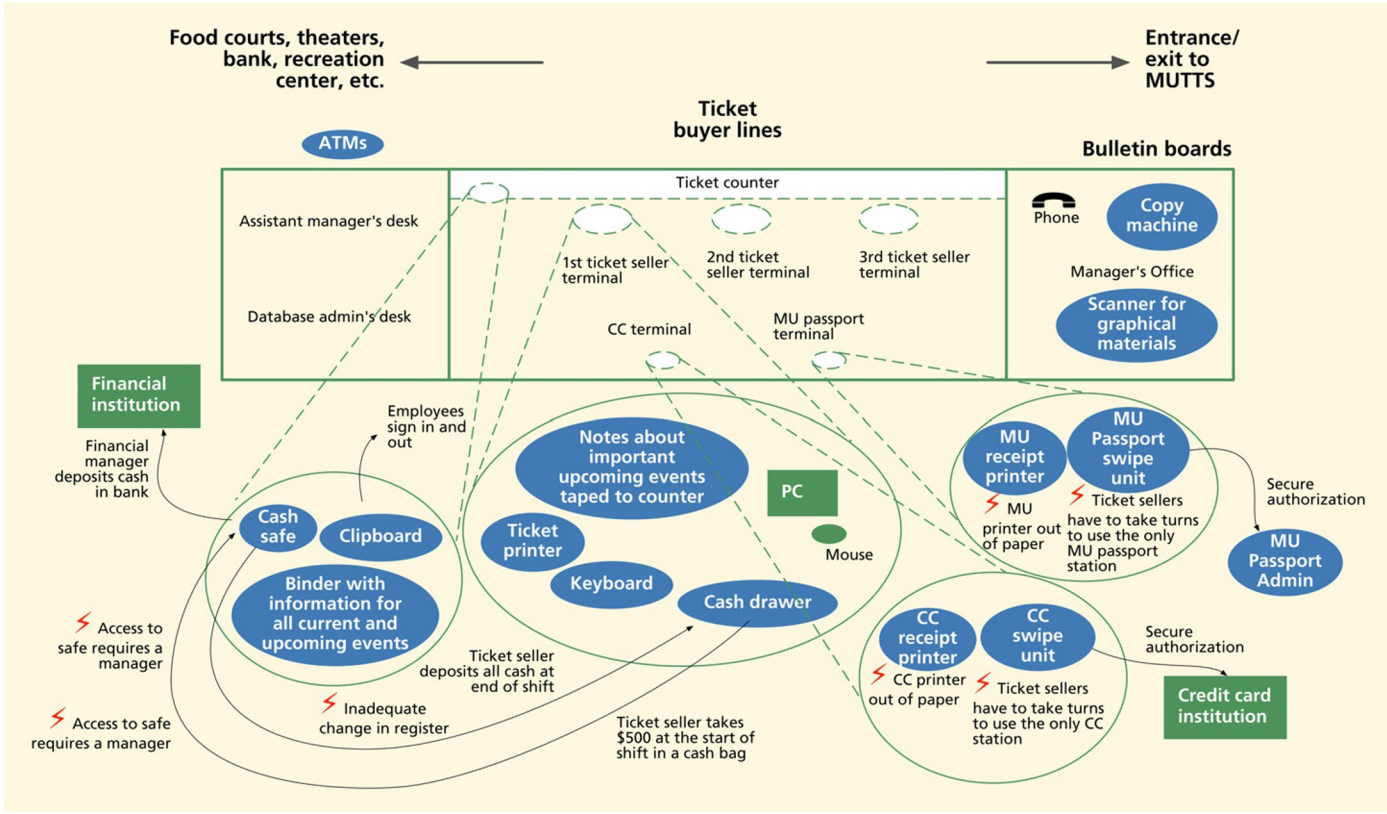

Fig. 9-14

A physical model for MUTTS.

material such as posters or advertisements for website content, presents barriers to usage: It is very slow and is not in a convenient location for all to share.

The ticket printers can also introduce barriers to workflow. Because they are specialized printers for ticket stock, when one goes down or runs out of paper or ink, the employees cannot just substitute another printer. They have to wait until the technician can get it serviced or, worse, until it can be sent out for service.

Information architecture

Design of information structures for organizing, storing, retrieving, displaying, manipulating, and sharing. Information architecture also includes designs for labeling, searching, and navigating information (Section 12.4.3).

9.10 INFORMATION ARCHITECTURE MODEL

If the data and information users need to access, store, display, and manipulate is complex, represent it with an information architecture model. What information do users use and interact with as part of their work and how is it structured?

Information architecture is usually fairly simple in the product perspective but could be quite complex in the enterprise system perspective. In projects where, for example, a database is central to work flow and/or data definitions and data relationships are crucial to understanding the work, information architecture can be intricate and essential to understand early.

In usage research analysis, as you encounter elemental data notes related to the structure of any system data or information involved in the work practice, merge them into the information architecture model.

An example is information about the fields and data types represented within personal contact information within a "contacts" application, such as for name, address, and phone number.

The TKS is a good example of a system centered on data objects such as events and tickets. Simple information structures can be represented as a list of information objects and their attributes, such as event name, event type, and event description.

The means for representing more complex structures such as a patient health record involve database schemas and entity-relationship diagrams. Database schemas, which can be huge, tie together usage and design, and are the basis for information displays and data field layouts in wireframes.

Also identify information that has to be "pervasive," that is, shared among users and devices over a larger ecology (Chapter 16). Pervasive information must have a consistent appearance and accessibility across multiple different work contexts.

Example: Information Architecture Modeling

Consider information about events in the MUTTS database. A general event record might have attributes like this:

- Event name.
- Event type.
- Event description.
- Range of dates event is occurring.
- Ticket costs:
  - Seat types and costs.
  - Reserved status.
- Venues:
  - Location.
  - Capacity.
- Directions to venues.
- Video trailers.
- Photos.
- Reviews.

Suppose you have to register each ticket buyer, especially if they pay by credit card. The ticket buyer might have attributes like these:

- Name.
- Address.
- Email address.
- Phone number.

There might also be a relationship between events and ticket buyers, including these attributes:

- Date of reservation.
- Seat number.
- Cost of ticket, as paid.

9.11 THE SOCIAL MODEL

Our social model is based on Beyer and Holtzblatt's (1988) cultural model. In practice, the social model is the least commonly used model, especially in agile environments. It is needed only when social and cultural interactions among the people involved in the work practice are complex and/or problematic.

The social model can be difficult to make because the information needed is seldom volunteered by users during data elicitation. A social model diagram can be difficult to understand because the arcs and labels of a graphical representation are too busy and too detailed; plus, all that structure isn't really necessary. It can be difficult to apply a social model to inform UX design because, technical design solutions for social problems can be elusive. We include this model mainly for completeness.

9.11.1 The Social Model Captures the Culture of the Shared Workplace

The social model captures the communal aspects of the users' organizational workplace as a social setting, and how they impact how things get done in the work environment, including:

Workers often have issues connected to:

- Work roles.
- Goals.
- How things get done in the work domain.
- Other workers.

9.11.2 Simplified Approach to the Social Model

In the first edition of this book, we spelled out a detailed and complicated way to make a social model as a graph with labeled nodes and arcs. Feedback from instructors and students revealed how difficult this model was to teach and learn. Feedback from UX professionals helped us realize that almost no one in this field would take the time and effort to make such a complicated model in a real project.

So, for those who want or need to make a social model, we have tried to reduce it to the most important aspects. It ended up being one case where a structured textual representation was simpler than a graphical one. See the example of a social model for MUTTS below.

9.11.3 Identify Active Entities

Active entities in a social model include all user work roles and can include any nonindividual agent or force that participates in, influences, or is impacted by the work practice, internal to or external to the immediate work environment.

In addition to the work roles and entities internal to the client organization, there are external roles that interact with work roles, including outside vendors,

customers, regulatory agencies, “the government,” “the market,” or “the competition.”

Example: List of social model entities for TKS

Our example of a social model for MUTTS starts with the roles. We identify the ticket seller and ticket buyer as the main ones, represented as list items at the highest level. You will almost always want to include the ambience and work domain as a nonhuman entity. The administrative supervisor, database administrator, and office manager are also shown in this list for TKS:

- Ticket seller.
- Ticket buyer.
- Ambience.
- Work domain.
- Administrative supervisor.
- Database administrator.
- Office manager.

9.11.4 Identify Kinds of Issues, Pressures, Worries, and Concerns

Categories of social model issues include matters such as:

- Concerns and worries.
- Influences exerted or felt.
- Job pressures.
- Issues affecting job performance.

More specific examples include:

- The overall flavor or feeling in the workplace.
- Organizational philosophy and culture.
- Workplace ambience and environmental factors.
- Professional and personal goals of workers.
- Political structure and realities.
- Thought processes, mindsets, policies, feelings, attitudes, and terminology that occur in the work environment.
- Legal requirements and regulations.
- Organizational policies.
- Subversive activities.

Personal and professional interrole influences. People in different work roles can influence people in other work roles on both a personal and a professional level. For example, the model may reflect plain old interpersonal or interrole frictions and animosities.

Power influences. There are many kinds of power within most organizations. Employees can "pull rank" based on official job titles. Also, people who proactively take on leadership roles can exert power. Influence also comes from the strength or authority a person exerts in a work role. In meetings, to whom does everyone listen the most? When the chips are down, who gets the job done, even it if means working outside the box?

Workplace ambience. The atmosphere of the workplace can have a psychological impact on users. For example, sometimes you might encounter a working environment that employees consider to be "toxic." If not addressed, this kind of social environment can lead to employee burnout, rebellion, and other counter-productive behavior.

Example: A Doctor's Office

It is possible to imagine a doctor's office as a stressful environment for employees. The general mood or work climate is rushed, chronically overbooked, and behind schedule. Emergencies and walk-ins add to the already high workload. The receptionist is under pressure to make appointments correctly, efficiently, and without errors. Everyone feels the constant background fear of mistakes and the potential of resulting lawsuits. Employees can't wait for the work day to end so they can escape to the relative peace of their homes (or the local pub).

Professional goals versus personal goals. Two different roles might view the same task from the perspective of different goals. For example, a manager might be concerned with capturing very complete documentation of each business transaction, whereas the person in the work role that has to compile the documentation may have as a goal to minimize the work involved. If we do not capture this secondary user goal in our analysis, we may miss an opportunity to streamline that task in design and the two goals may remain in conflict.

Subversion among the ranks. Be sure to include in your social model how people think and act negatively in response to dissatisfaction. Is the watercooler or the break room the center of subversive coordination? Is

subversion or passive-aggressive behavior a common answer to power and authority? How strong is the "whistleblower" mentality? Does the organization thrive on a culture of guerilla activity?

9.11.5 Add Concerns and Influences to the Social Model List

In usage research analysis, as you encounter elemental data notes related to social concerns in the workplace, merge them into the evolving social model. Add the corresponding active entities to the list and, indented under each entity, add any related concerns and issues.

Example: A Social Model for MUTTS

Here is an excerpt of what a social model might look like for MUTTS:

- Ticket buyer:
  - Feels pressure:
    - From those in line to get done fast.
    - To get good seats for popular events.
  - Worries about:
    - The system completing the transactions correctly.
    - Losing the personal service I like so much if it's changed to a kiosk.
    - Losing my tickets or my money if the system has an error.
    - Losing money due to the no-refund policy.
  - Experiences barriers to work in the form of:
    - Noise and distraction of public location interfere with thinking and decision-making.
    - Can't see all the choices upfront.
- Ticket seller:
  - Feels pressure:
    - Pushback from users about the no-refund policy.
    - To please customers.
    - Peak periods mean fast pace and hard work.
  - Worries about:
    - Complaints about service that can ruin job reviews.
  - Is subject to external influences of:
    - The administrative supervisor, who shows up occasionally and causes stress.
    - The administrative supervisor, who is not familiar with daily operations and can impose unrealistic expectations.

- Administrative supervisor:
  - Worries about not generating enough revenue.
  - Would like to increase revenue with other merchandise sales, including over-the-counter commodities such as candy, gum, and aspirin plus merchandise souvenirs, T-shirts, hats, and banners.
- Database administrator:
  - Pressure to:
    - Maintain data integrity.
    - Keep system up and running continuously.
  - Experiences barriers to work in the form of:
    - Event manager's phone, the constant ringing of which makes it hard to concentrate, especially when something is going wrong.

Exercise 9.8: A Social Model for Your Product or System

Goal: Get a little practice in making a social model diagram.
Activities: Identify active entities, such as work roles, and represent as bullets in a list. Include:

- Groups and subgroups of roles and external roles that interact with work roles.
- System-related roles, such as a central database.
- Workplace ambience and its pressures and influences.

Next identify:

- Concerns and perspectives to be represented subbullets in the list.
- Social relationships, such as influences between entities, to be represented as subbullet items under appropriate entities.
- Barriers, or potential barriers, in relationships between entities to be represented as red bolts of lightning ().

Deliverables: One social model diagram for your system with as much detail as feasible.
Schedule: This could take a couple of hours.

Exercise 9.9: A Social Model for Smartphone Usage

Sketch out an annotated social model for the use of an iPhone or similar smartphone by you and your friends. We're giving you lots of leeway here; make the most of it.

9.12 HYBRID MODELS

While we describe each usage research data model separately for clarity, in practice models are often combined for efficiency. For example, because the flow model includes at least the major user work roles, you don't necessarily need a separate user work role model. You can just label the user work role nodes of the flow model with the user work role names and annotate them with user class characteristics. Later, you can even label flow model nodes with the corresponding personas.

The goal of modeling is to represent a useful perspective or perspectives about the work domain for the design situation. Purity of each model is not the goal. Do what it takes to capture what you learn about the work domain. For example, if the work practice centers around a physical space, like a ticket office, combine the physical model and flow model into a hybrid model.

9.13 MODEL CONSOLIDATION

In a large project, if you construct your models with multiple subteams working in parallel, you will get multiple models of the same type. Now is the time to consolidate the model versions by merging them into one model. The key idea is to induce generalizations, that is, a bottom-up process to build a general model from the important pieces of specific data.

As an example, start with representations of single-user stories of task steps in the existing work practice. Merge the description of essentially the same task created with data from several users, and factor out the differences in details. The result is a more abstract or more general representation of the interaction, representing how most or all users do the task.

When flow modeling is done by different subteams, each model can be different. The same work role might get modeled in different ways, yielding different work role descriptions and work role names. Because these various versions of the flow model are about the same workflow, they can be consolidated essentially by merging them.

Example: Flow Model Consolidation for MUTTS

Figs. 9-15, 9-16, and 9-17 are partial flow models constructed by groups who observed and interviewed different people of the overall organization and work practice.

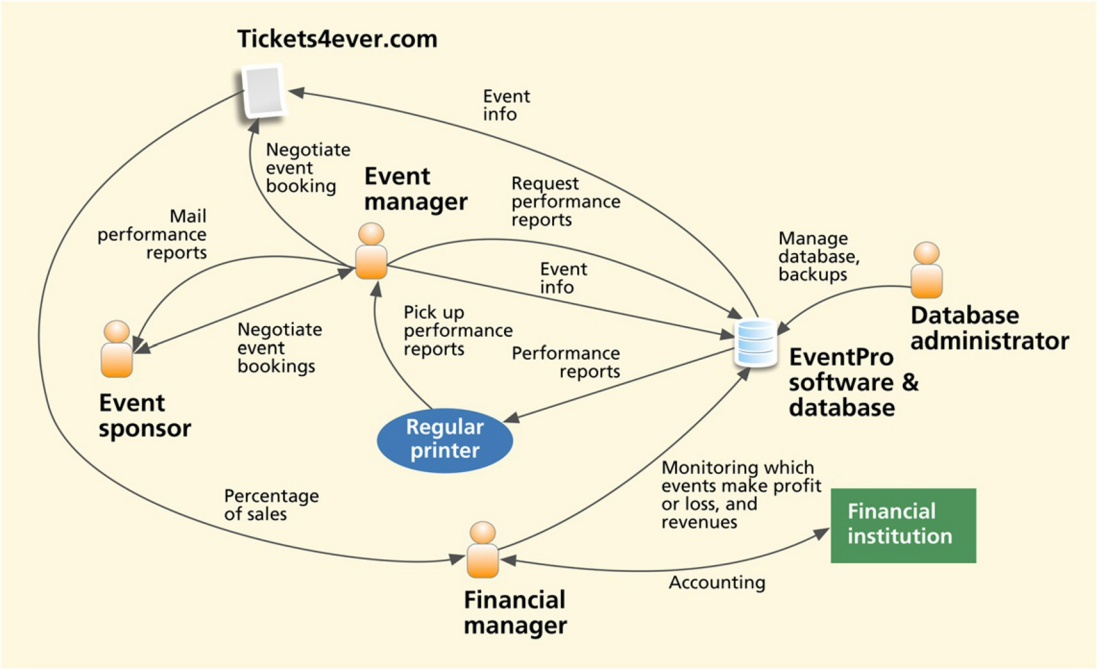

Fig. 9-15

Flow model from a group who observed and interviewed the event manager, event sponsors, the financial manager, and the database administrator.

Look back at Fig. 9-4 to see how the three parts of the overall flow model came together in model consolidation.

9.14 WRAP UP

9.14.1 Barrier Summaries Across All Models

Many of the models tell partial stories from different perspectives, but no one model highlights all the barriers discovered in usage research data elicitation and analysis. Yet it is the barriers to work practice and user performance that most directly inform design ideas for the new system. So it can be helpful and informative to extract barrier-related information from the models and summarize the barriers in one place.

Barriers to usage are of special interest because they point out where users have difficulties in the work practice. Anything that impedes user activities,

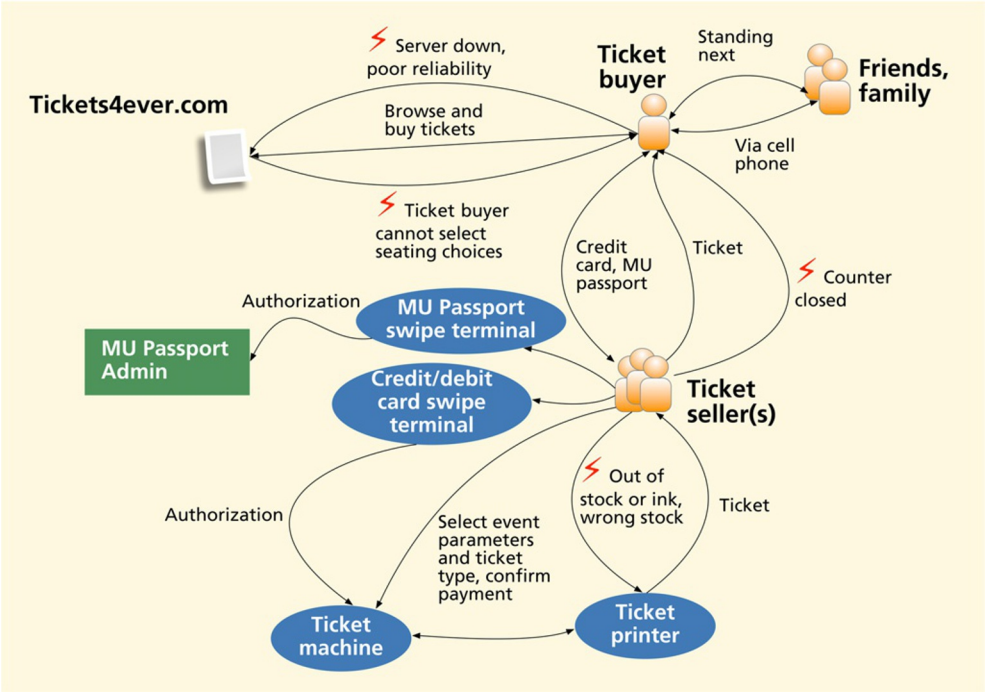

Fig. 9-16

Flow model from a group who mainly observed and interviewed ticket buyers and ticket sellers.

interrupts workflow or communications, or interferes with the performance of work responsibilities is a barrier to the work practice. Because these barriers also represent key opportunities for improvement in the design, it is useful to collect them all into a single summary of barriers.

9.14.2 Post Data Models in Your UX Design Studio

Make your models visual in sketches and posters prominently displayed as part of the immersion in your design studio. Assemble your artifacts into an organized display.

Considerations about posting in your UX design studio.

- In early stages, it might be best to draw your diagrams and figures on whiteboards, where it is easy to discuss and update them.

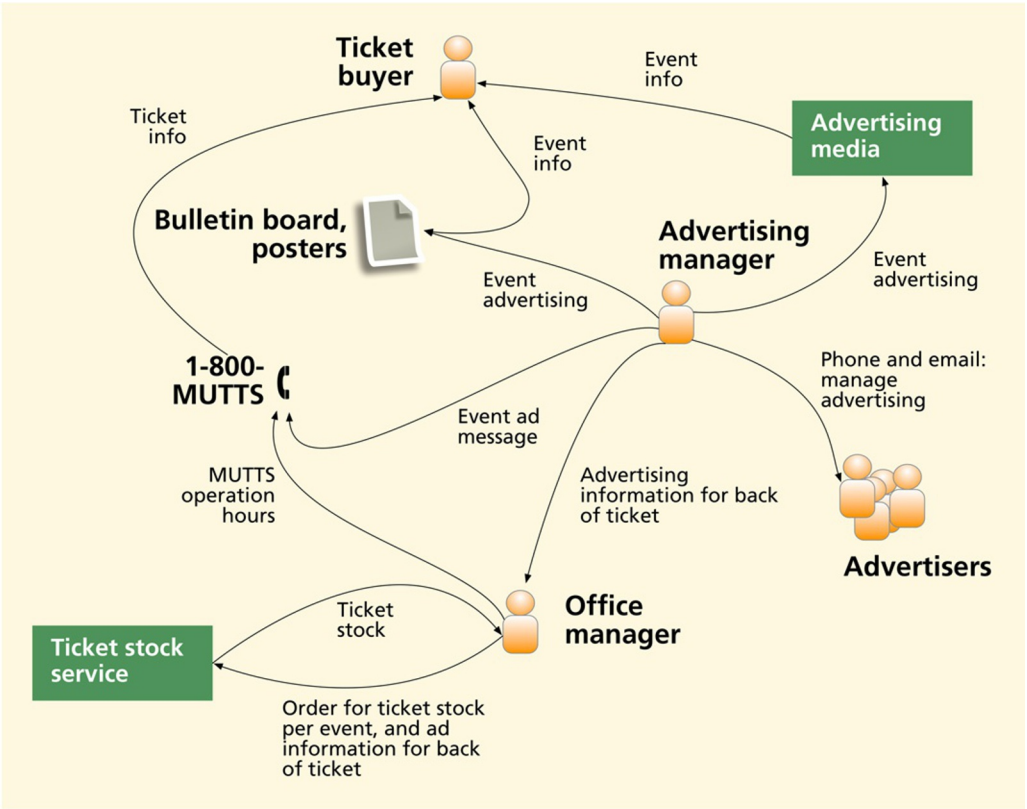

Fig. 9-17

Flow model from a group who observed and interviewed the office manager, the advertising manager, and external advertisers.

- Later, you can transfer the diagrams to nicely printed posters taped to the wall.
- Assemble your artifacts into an organized collection on a central work table.
- At some point in projects of any realistic size, you will end up with more materials to post than you have space for posting, so:
  - Be judicious; trim out the less important items.
  - Post artifacts and diagrams that will have importance throughout the entire project.
  - Diagrams, images, and sketches that you will use less frequently could be kept in your laptop and, when needed, shown via a projector.

- Beyond this, larger amounts of shared information can be kept in sharable files (e.g., in Dropbox or Google Drive).
- Retire postings for which the usefulness has diminished over time.
- Remember, anything posted carries the task of keeping it up to date.
- As you move to designs, almost all the early postings give way to annotated wireframe sketches, which become the language of communication.
- Posted wireframe sketches also support design walkthroughs with stakeholders.